Translating Scheduled Transfer Protocol Between HIPPI-6400 and HIPPI-FP

(Normative Annex)

1.0 Overview

This annex describes the translation of Scheduled Transfer Protocol between a HIPPI-6400 host which is implementing Scheduled Transfers and a HIPPI-800 host which is also implementing Scheduled Transfers. There are two alternatives for this translation. The first alternative, Pass Through Translation, passes all Scheduled Transfer protocol information between the HIPPI-6400 host and HIPPI-800 host with no modification. The second alternative, Fragmenting Translation, divides large HIPPI-800 packets into multiple smaller HIPPI-6400 blocks and vice versa. In both cases the HIPPI-800 message format uses a Mac header which is identical in format and content to the HIPPI-6400 Mac header (including the IEEE header).

2.0 Pass Through Translation

In Pass Through Translation all Scheduled Transfer protocol information, as well as the Mac header, shall be passed between the HIPPI-6400 host and HIPPI-800 host with no modification. On the 800 side, these shall be of the form shown in Figure 2-1. The I-Field shall contain the proper HIPPI-800 logical addresses. The FP Header shall contain the proper lengths and the ULP-ID shall be x'08'. The D1 Area shall contain valid MAC and Schedule Transfer headers. The D2 Area, if present, shall contain the payload being transferred.

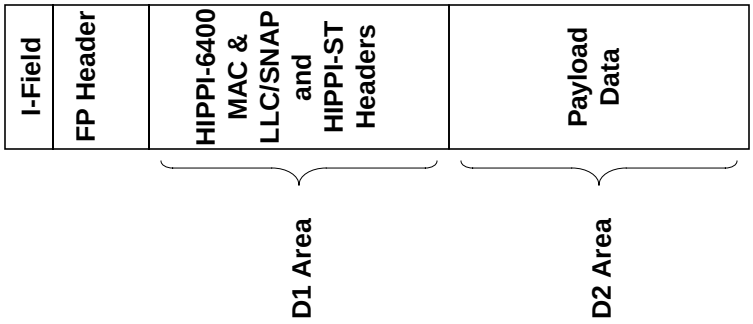
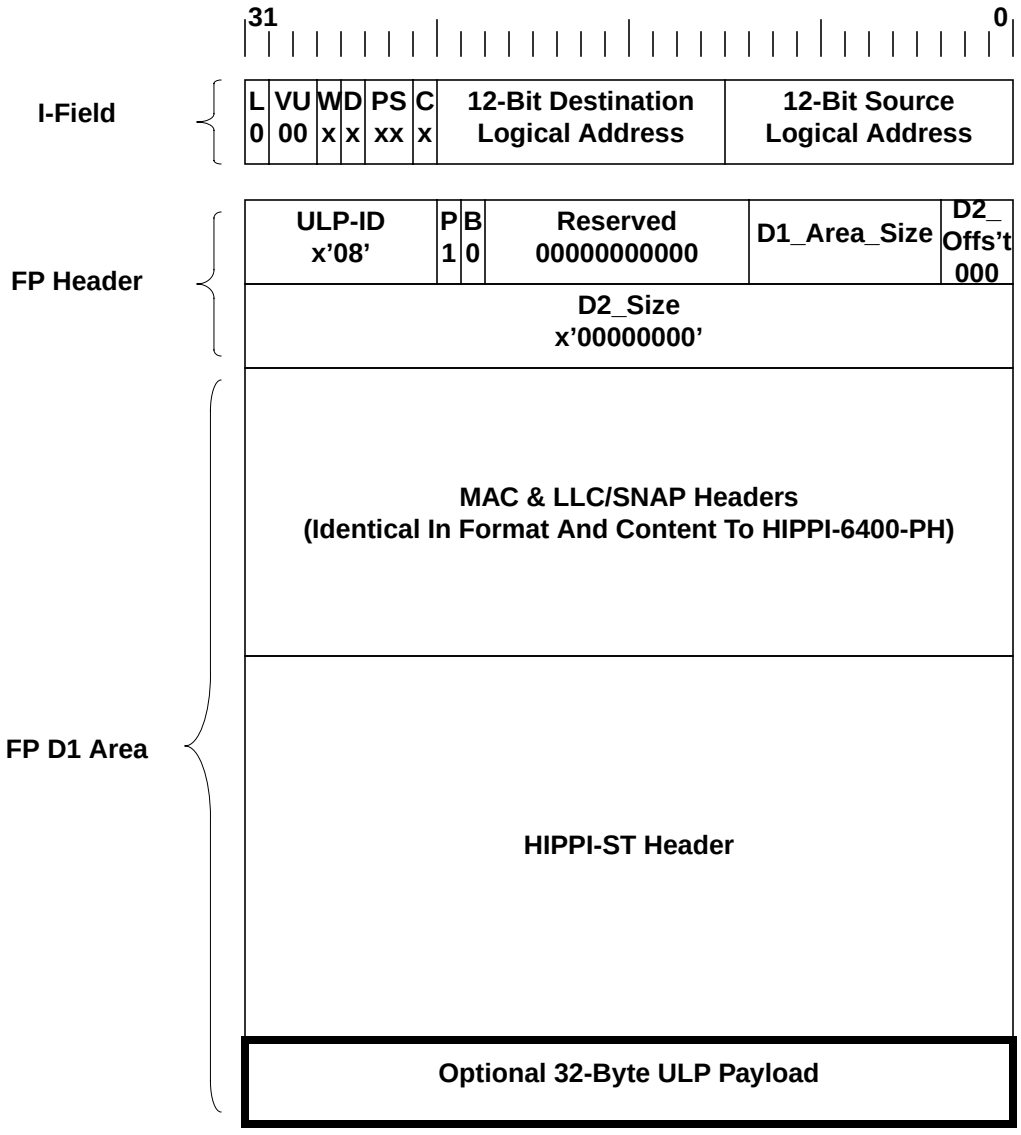


Figure 2-1 Pass Through Scheduled Transfer Message on HIPPI-800

2.1 HIPPI-800 To HIPPI-6400 Control Message

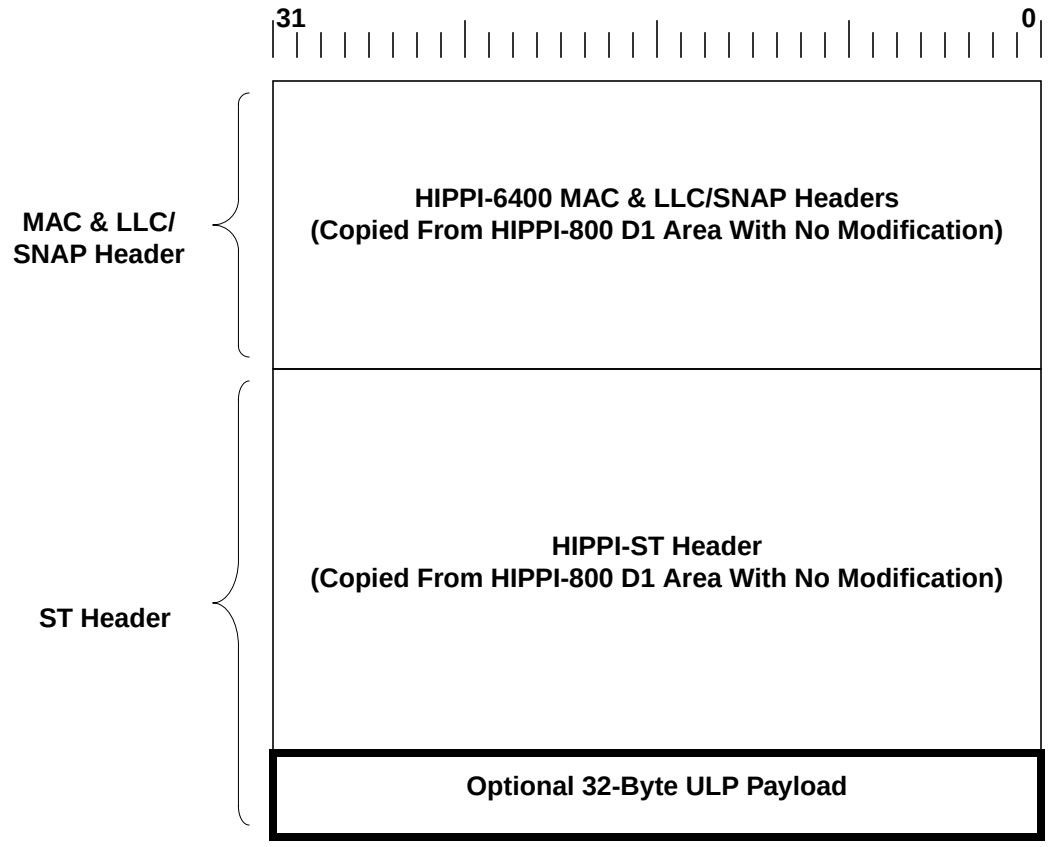
Figure 2-2 shows the HIPPI-800 format for a Scheduled Transfer control message (non-data). Figure 2-3 shows the resulting HIPPI-6400 format for the Scheduled Transfer control message (non-data).

Figure 2-2 HIPPI-800 To HIPPI-6400 Control Message
HIPPI-800 Format
(Pass Through Translation)



1. 12-Bit Destination Logical Address is 12 LSB's of HIPPI-6400 destination host's 48-bit logical address
2. D1_Area_Size specified as number of 64-bit words

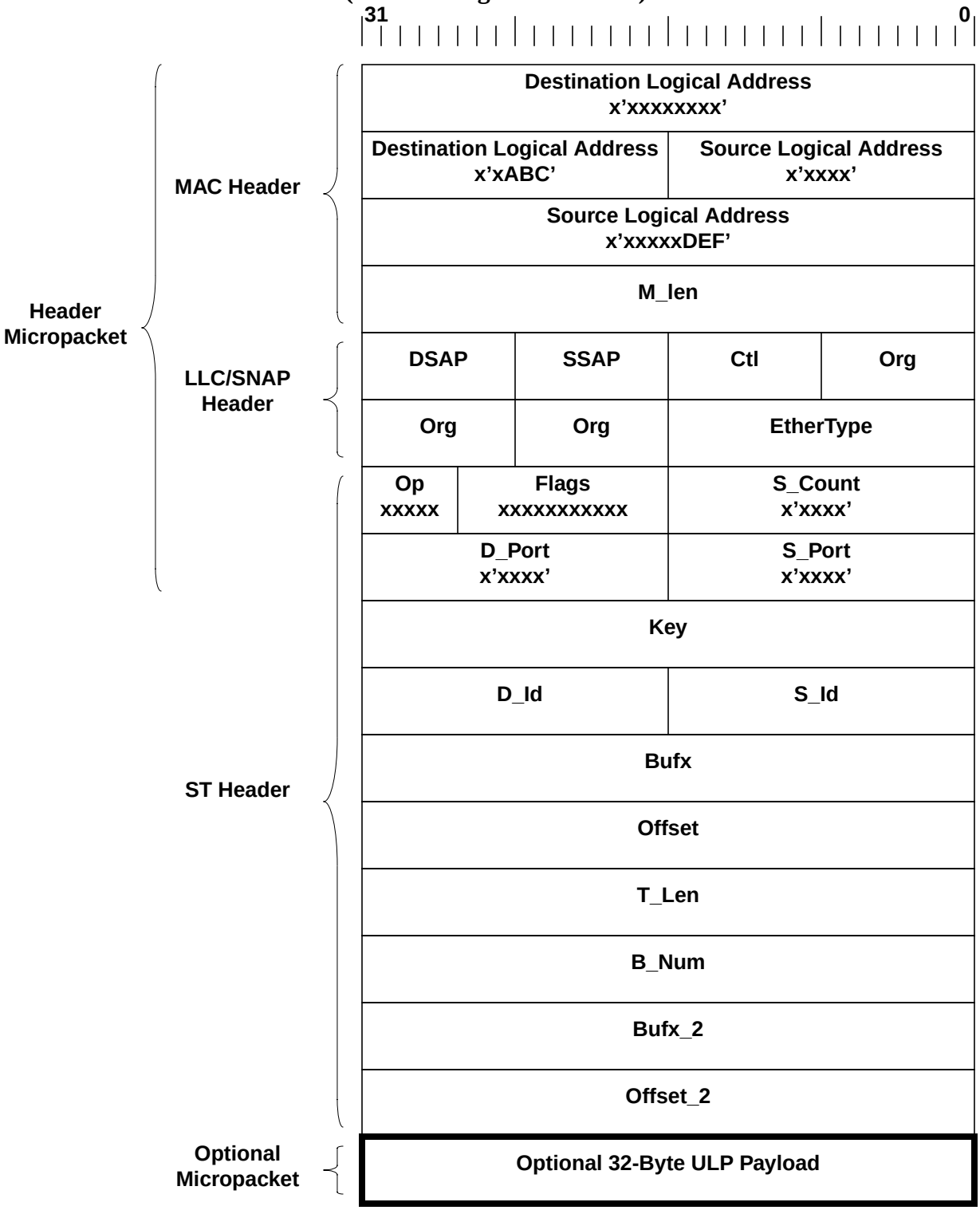
Figure 2-3 HIPPI-800 To HIPPI-6400 Control Message
HIPPI-6400 Format
(Pass Through Translation)



2.2 HIPPI-6400 To HIPPI-800 Control Message

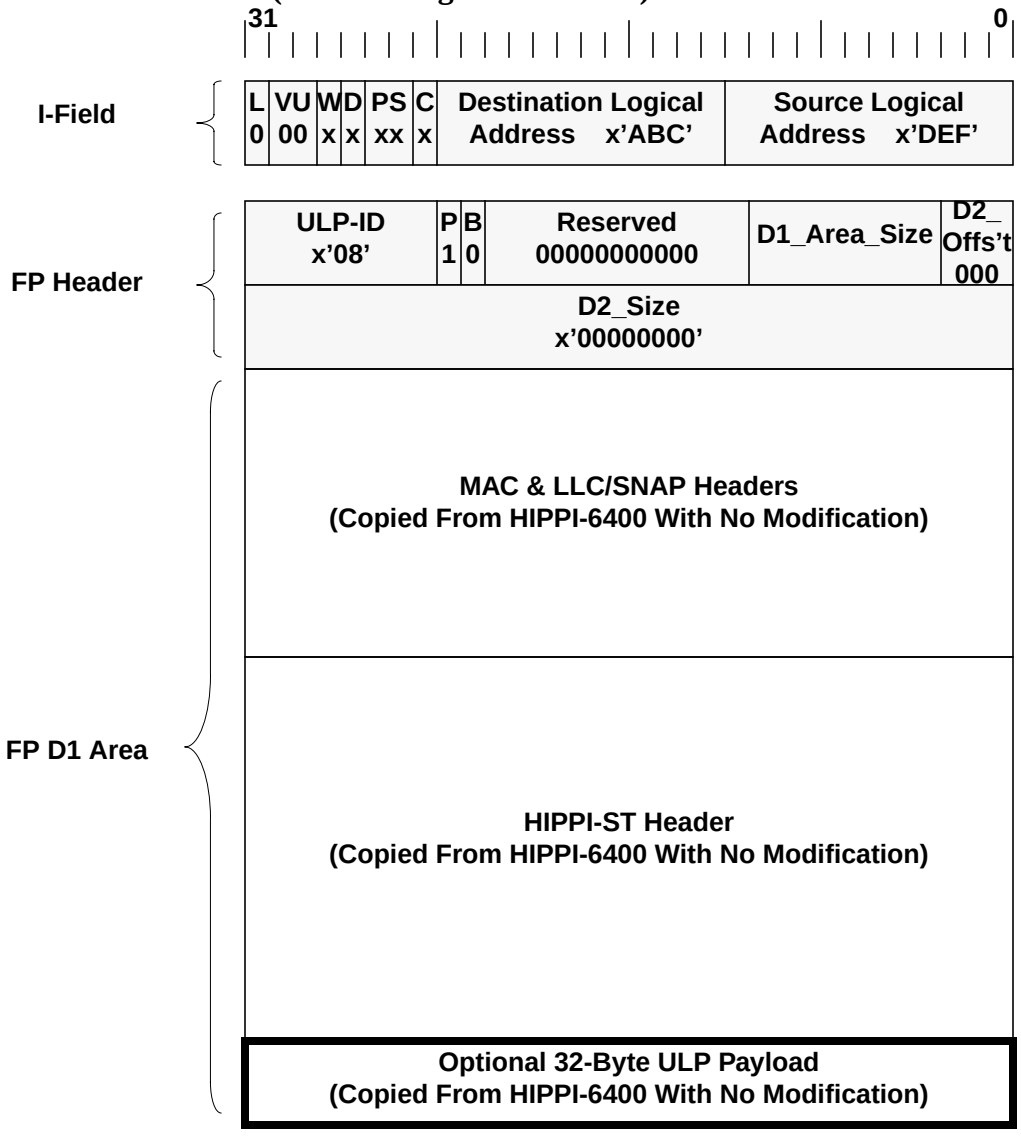
Figure 2-4 shows the HIPPI-6400 format for a Scheduled Transfer control message (non-data) which originated at the HIPPI-6400 host. Figure 2-5 shows the resulting HIPPI-800 formatted message.

Figure 2-4 HIPPI-6400 To HIPPI-800 Control Message
HIPPI-6400 Format
(Pass Through Translation)



ABC = HIPPI-800 destination address
DEF = HIPPI-6400 source address

Figure 2-5 HIPPI-6400 To HIPPI-800 Control Message HIPPI-800 Format (Pass Through Translation)

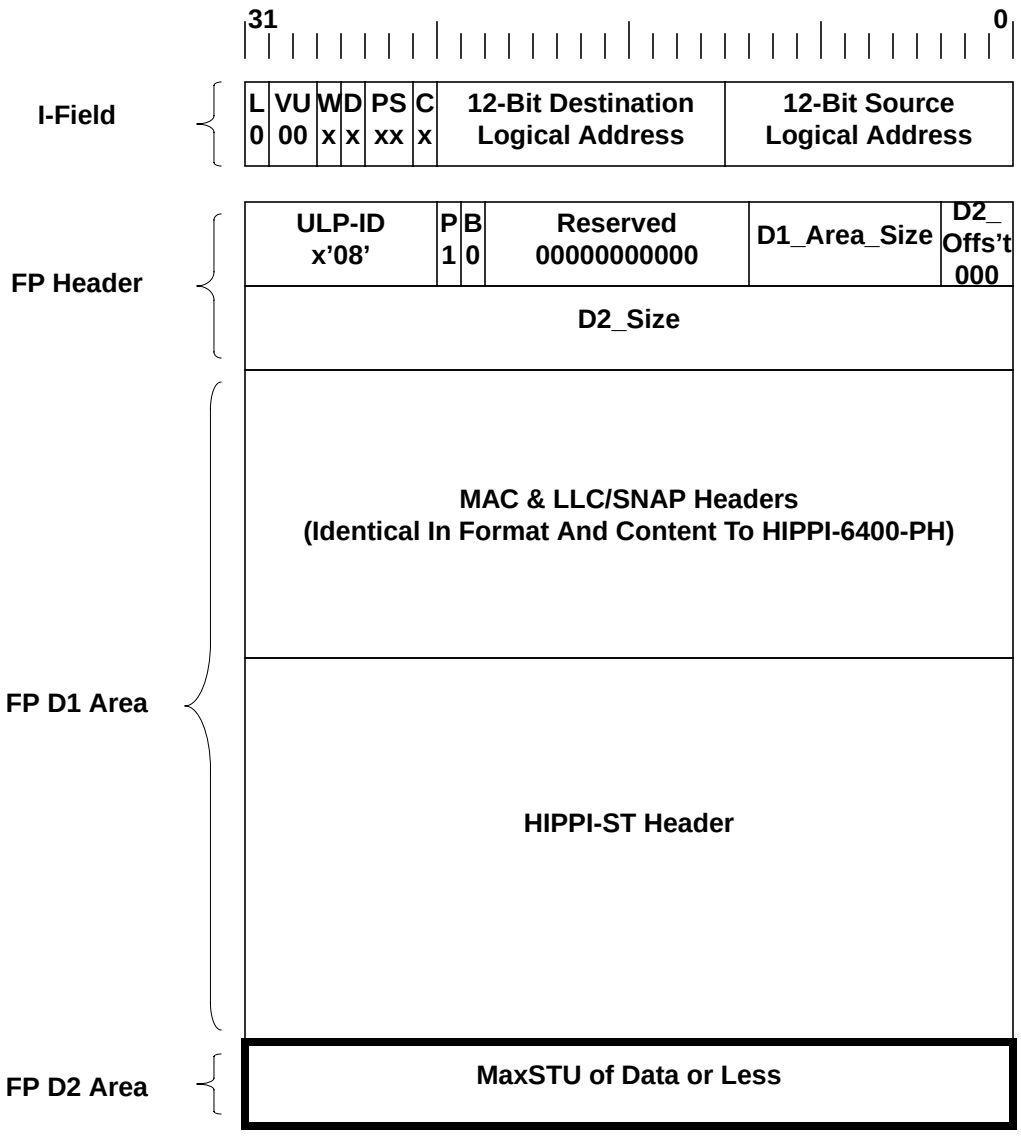


1. 12-Bit Logical Addresses are 12 LSB's of HIPPI-6400 48-bit logical address
2. D1_Area_Size = (HIPPI-6400 MAC Header M_Len field plus 16 bytes)/8 (D1_Area_Size specified as number of 64-bit words)

2.3 HIPPI-800 To HIPPI-6400 Data Message

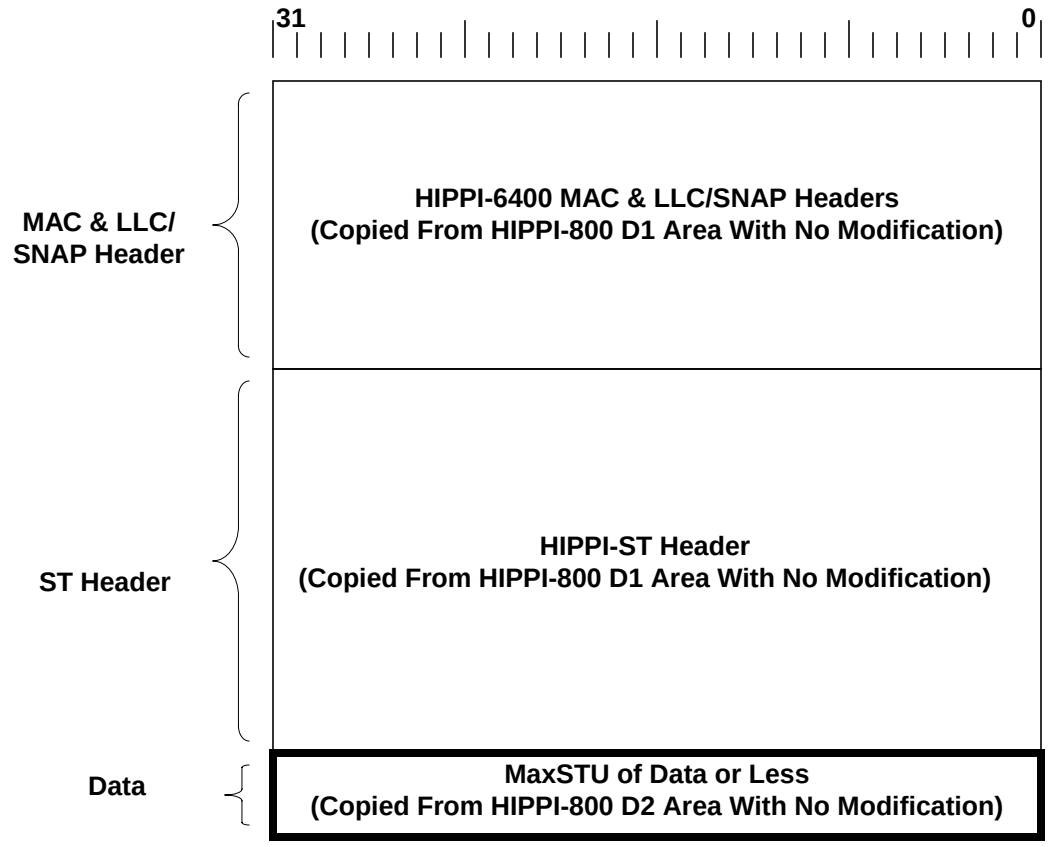
Figure 2-6 shows the HIPPI-800 format for a Scheduled Transfer data message which originated at the HIPPI-800 host. Figure 2-7 shows the resulting HIPPI-6400 formatted message.

Figure 2-6 HIPPI-800 To HIPPI-6400 Data Message
HIPPI-800 Format
(Pass Through Translation)



1. 12-Bit Destination Logical Address is 12 LSB's of HIPPI-6400 destination host's 48-bit logical address
2. D1_Area_Size specified as number of 64-bit words

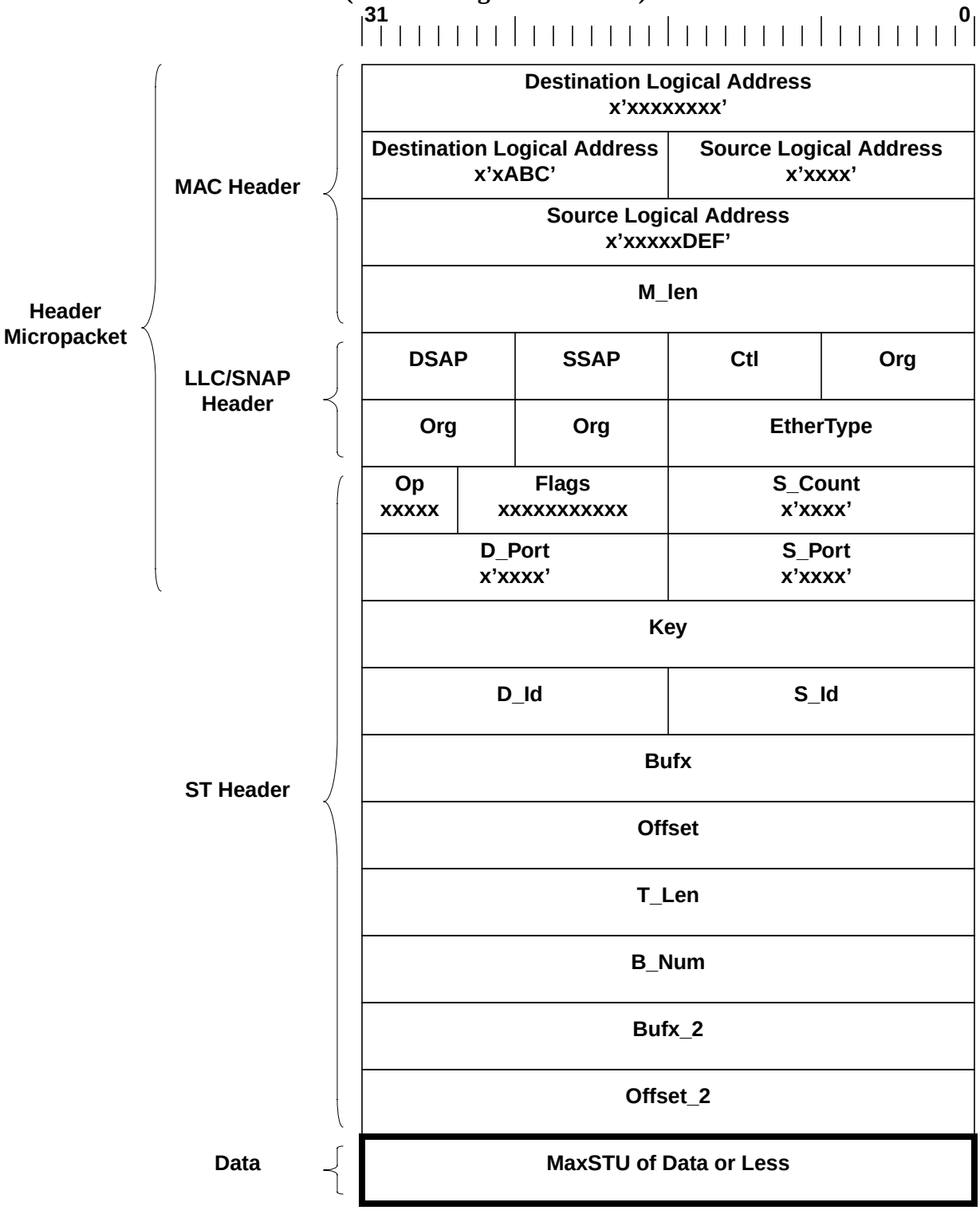
Figure 2-7 HIPPI-800 To HIPPI-6400 Data Message
HIPPI-6400 Format
(Pass Through Translation)



2.4 HIPPI-6400 To HIPPI-800 Data Message

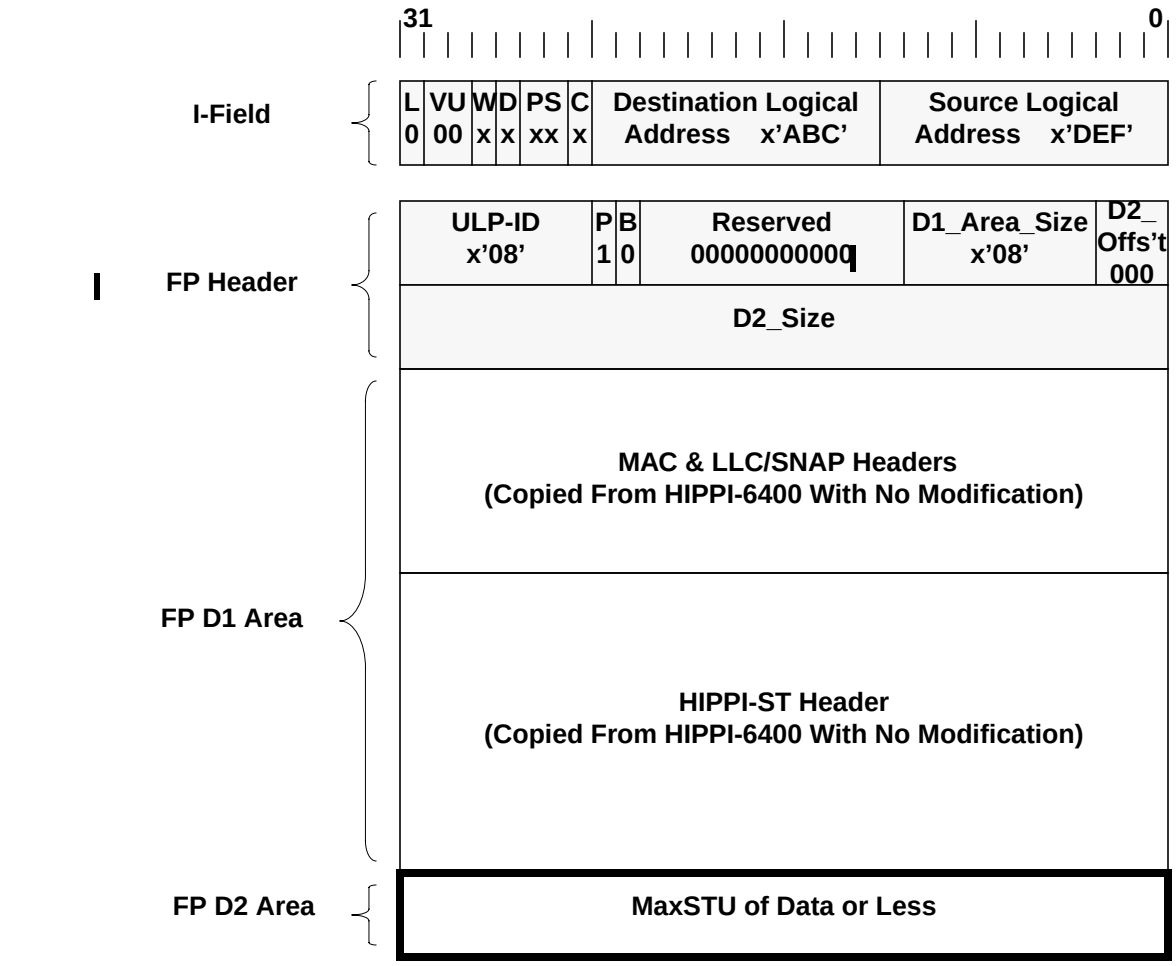
Figure 2-8 shows the HIPPI-6400 format for a Scheduled Transfer data message which originated at the HIPPI-6400 host. Figure 2-9 shows the resulting HIPPI-800 formatted message.

Figure 2-8 HIPPI-6400 To HIPPI-800 Data Message
HIPPI-6400 Format
(Pass Through Translation)



ABC = HIPPI-800 destination address
DEF = HIPPI-6400 source address

Figure 2-9 HIPPI-6400 To HIPPI-800 Data Message HIPPI-800 Format (Pass Through Translation)



1. 12-Bit Logical Addresses are 12 LSB's of HIPPI-6400 48-bit logical address
2. D2_Size = HIPPI-6400 M_Len-48

3.0 Fragmenting Translation

In fragmenting translation large HIPPI-800 packets are divided into multiple smaller HIPPI-6400 blocks and vice versa. All control messages shall be passed through the translator unmodified with the exception of the CTS generated by the HIPPI-800 host.

The translator shall accept a single CTS from the HIPPI-800 host then create and send multiple CTS operations to the HIPPI-6400 host (multiple CTS operations are sent only if payload is larger than MaxSTU). STUs received from the HIPPI-6400 host in response to the multiple CTS operations shall be combined into one large packet and forwarded to the HIPPI-800 host.

Packets received from the HIPPI-800 host which exceed the MaxSTU size shall be fragmented into multiple STUs and forwarded to the HIPPI-6400 host.

3.1 Control Message Format

All control messages except CTS from the HIPPI-800 host shall be of the same format described in paragraph 2.0.

3.1.1 HIPPI-800 to HIPPI-6400 CTS Operation Processing

A CTS operation originating in the HIPPI-800 host shall be of the same format described above with the addition of a Replicate value in the first four bytes of the optional ULP payload field and Return_D2_Size value in the second four bytes of this field. The translator shall use the Replicate value to determine how many HIPPI-6400 blocks the following HIPPI-800 block is to be fragmented into. The translator shall send the designated number of CTS operations to the HIPPI-6400 host, pacing the operations as required to maintain data flow and protect buffer space. A value of zero shall indicate no replication. A value of 1 shall indicate that two CTS operations are to be sent, etc. The Return_D2_Size value is the value to be inserted in the D2_Size field in the FP header of the HIPPI-800 packet which will be returned in response to this CTS. The Replicate and Return_D2_Size fields shall be zeros in the CTS operations sent to the HIPPI-6400 host.

3.2 HIPPI-6400 to HIPPI-800 DATA Messages Processing

Multiple STUs sent from the HIPPI-6400 host to the HIPPI-800 host in response to a fragmented CTS shall be combined into one HIPPI-800 packet. The translator shall use the MAC and Schedule headers from the first HIPPI-6400 STU to create an I-Field, FP header, and D1 region. The I-Field shall contain logical addresses equal to the LSB 12 bits of the HIPPI-6400 MAC addresses. The FP Header shall have a D2_Size value equal to the Return_D2_Size value received in the CTS operation from the HIPPI-800 host. The ULP-ID shall be x'08'. The D1 Area shall be a copy of the HIPPI-6400 MAC and Schedule Transfer headers, with S_count set to zero and the MAC length set to the same value as the D2 length plus 48 bytes. The D2 shall contain the STU payload copied from all of the HIPPI-6400 STUs (number of HIPPI-6400 STUs determined by Replicate value in HIPPI-800 CTS operation).

3.3 HIPPI-800 to HIPPI-6400 DATA Message Processing

Data sent from the HIPPI-800 host to the HIPPI-6400 host in a single HIPPI-800 packet shall be fragmented into multiple HIPPI-6400 STUs if needed. The translator shall use the MAC and Schedule header contained in the D1 area of the HIPPI-800 packet as the headers for the first HIPPI-6400 STU. The translator shall use this same information to repetitively create the proper MAC and Schedule header for subsequent HIPPI-6400 STUs (number of STUs determined by the D2_Size and MaxSTU size) by decrementing M_count and incrementing Offset by MaxSTU. Since the HIPPI-800 host has already scheduled this entire block of STUs before sending them to the translator, the translator does not need to do any handshaking at all in this direction. The last STU may be less than the MaxSTU size.

3.4 Message Processing Summary

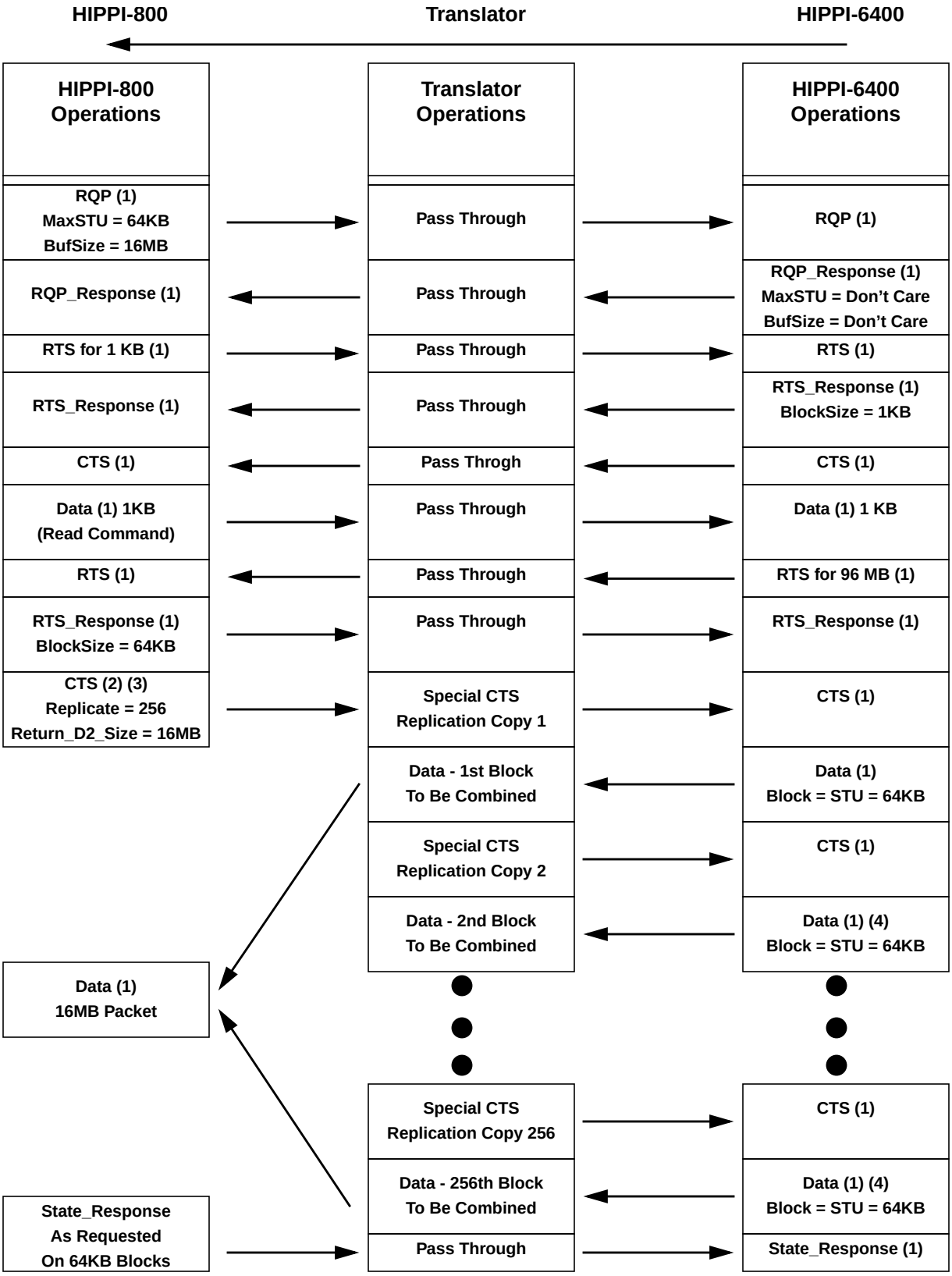
Table 3-I summarizes the processing performed by the translator for each Scheduled Transfer operation.

Table 3-I Fragmenting Translation Message Processing By Translator

Operation	Host Originating Message	
	HIPPI-800	HIPPI-6400
Request_Port (RQP)	Pass Through	Pass Through
RQP_Response	Pass Through	Pass Through
Port_Teardown	Pass Through	Pass Through
Port_Teardown_ACK	Pass Through	Pass Through
Port_Teardown_Complete	Pass Through	Pass Through
Req_To_Send (RTS)	Pass Through	Pass Through
RTS_Response	Pass Through	Pass Through
Clear_To_Send (CTS)	Replicate	Pass Through
Req_To_Receive (RTR)	Pass Through	Pass Through
DATA	Fragment	Combine
Request_State	Pass Through	Pass Through
State_Response	Pass Through	Pass Through
End	Pass Through	Pass Through
End_ACK	Pass Through	Pass Through

3.5 Example Data Flows

Figure 3-1 shows Message flow for a 96 MB read from a HIPPI-6400 server to a HIPPI-800 client.Figure 3-2 shows Message flow for a 96 MB write from a HIPPI-800 client to a HIPPI-6400 server.



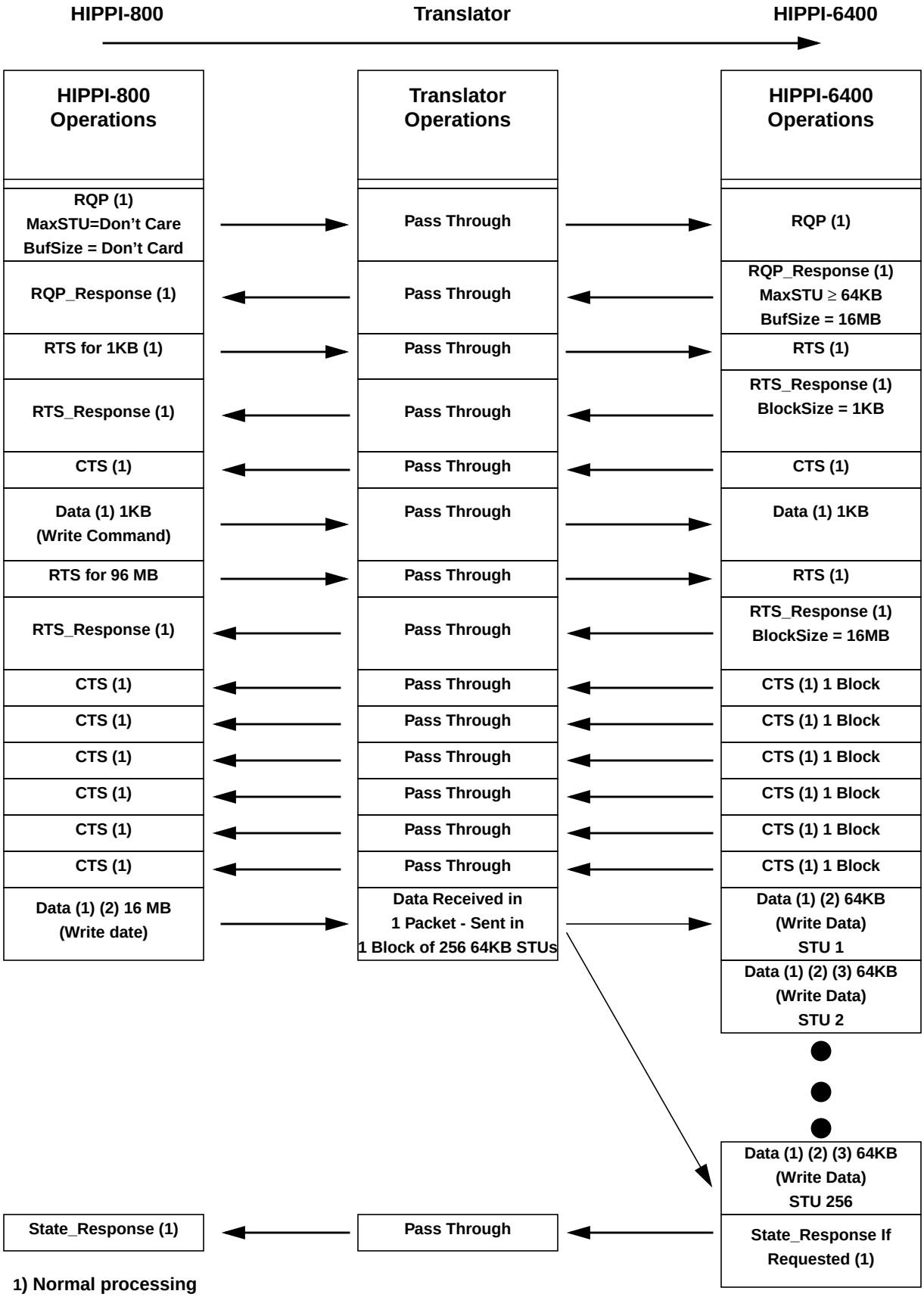
1) Normal processing

2) Special processing-This CTS represents authority for 256 CTS operations at the translator.

3) This CTS and subsequent operations are sent on 6 HIPPI-800 ports. Only one of these ports is shown.

4) Bufx and Offset need not change since the header will be discarded at translator.

Figure 3-1 96 MByte Read Operation Data Flow



1) Normal processing

2) Blocks are sent on 6 different HIPPI-800 channels for striping. Only one channel is shown in diagram.

3) Bufx is not modified, Offset increments by 64KB

Figure 3-2 96 MByte Write Operation Data Flow

3.6 HIPPI-800 To HIPPI-6400 CTS Operation

Figure 3-3 shows the HIPPI-800 format for a Scheduled Transfer CTS operation which originated at the HIPPI-800 host. Figure 3-4 shows the resulting HIPPI-6400 formatted CTS operation.

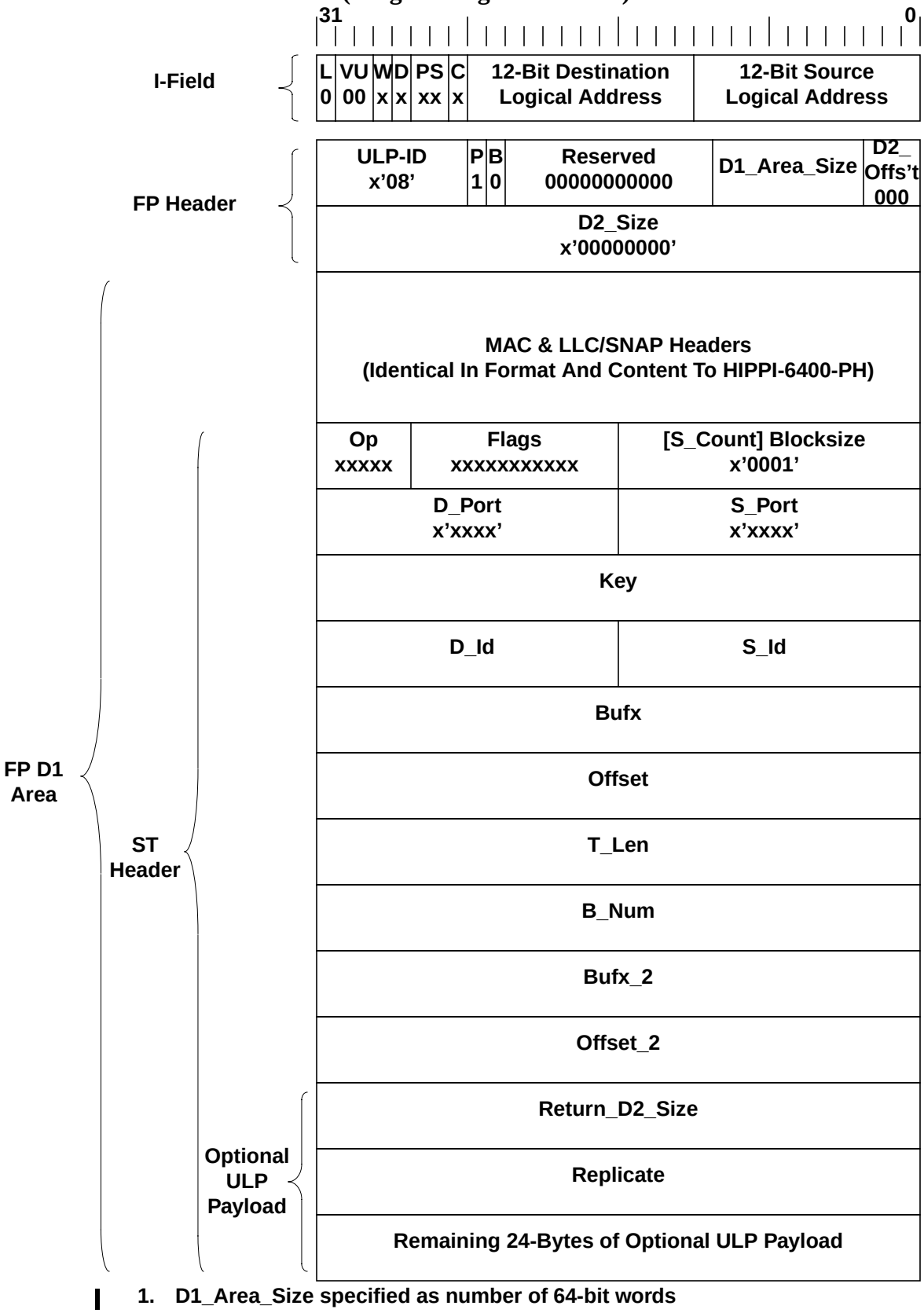
Notes:

1. 12-Bit Destination Logical Address is 12 LSB's of HIPPI-6400 destination host's 48-bit address

2. Replicate value carried in Offset_2 is number of blocks authorized by this CTS operation

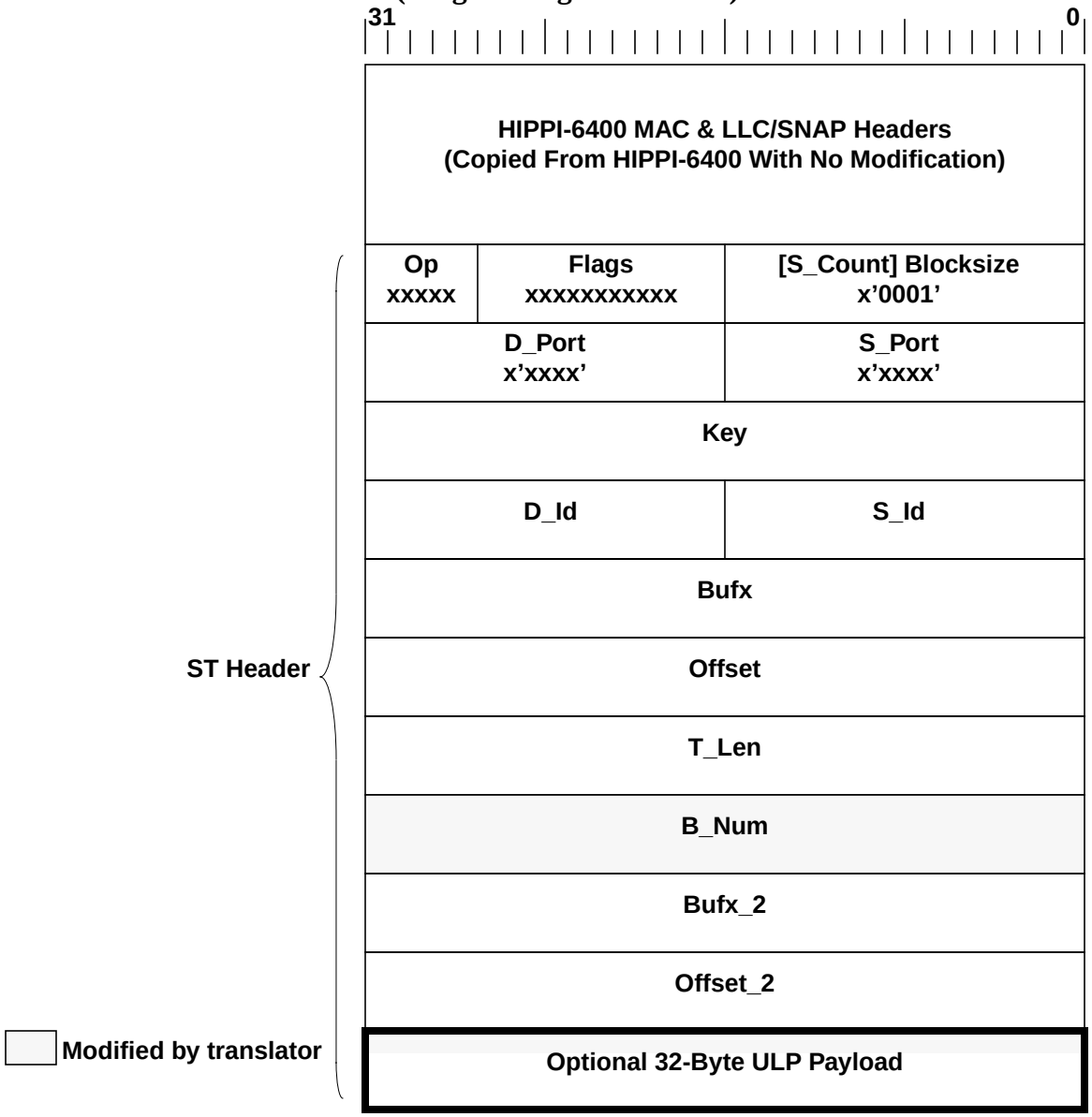
3. Return_D2_Size value is copied into D2_Size field in FP header of returned HIPPI-800 packet

Figure 3-3 HIPPI-800 To HIPPI-6400 CTS Message
HIPPI-800 Format
(Fragmenting Translation)



1. D1_Area_Size specified as number of 64-bit words

Figure 3-4 HIPPI-800 To HIPPI-6400 CTS Message HIPPI-6400 Format (Fragmenting Translation)

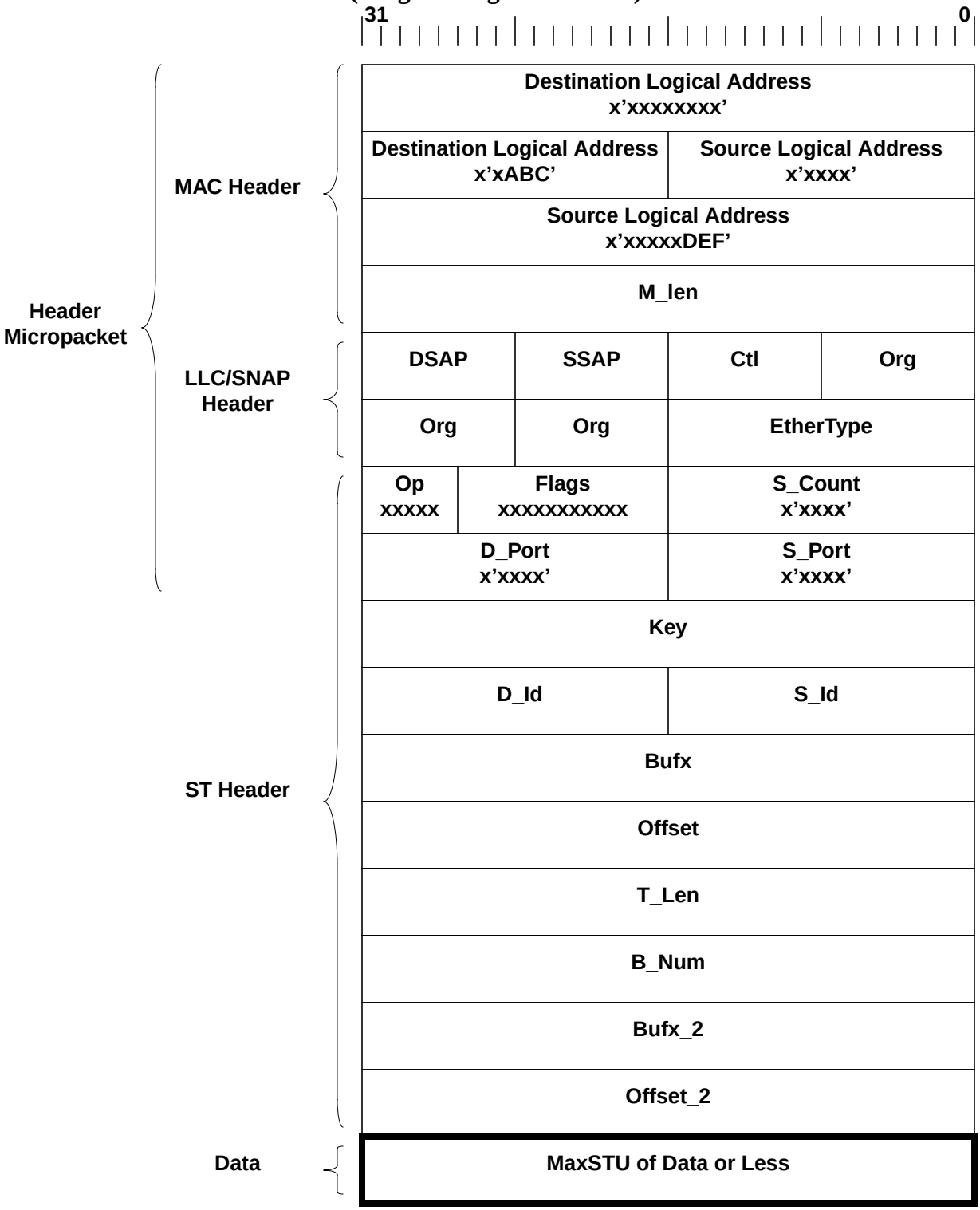


1. MAC & LLC/SNAP headers and ST header are stored and duplicated for subsequent CTS commands to HIPPI-6400 host (first set sent unmodified)
2. Correct addresses contained in original MAC header from HIPPI-800 D1 Area
3. B_Num starts at the value in the ST header from HIPPI-800 D1 Area and increments for each CTS sent to HIPPI-6400 host
4. Offset starts at value in ST header from HIPPI-800 D1 Area and doesn't change (this will be copied into header of first data STU, headers from subsequent CTS operations will be stripped and discarded)
5. First 8 bytes of optional payload are changed to zeros

3.7 HIPPI-6400 To HIPPI-800 Data Message

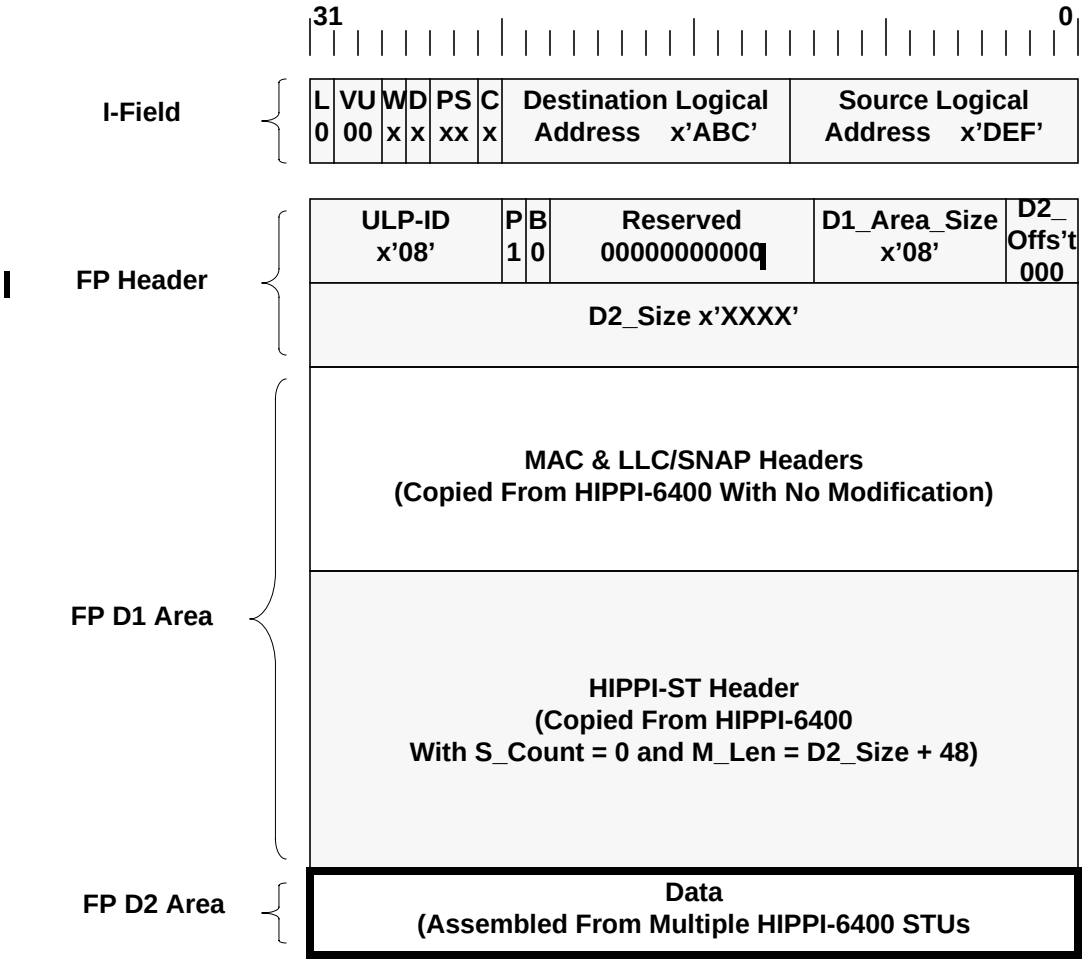
Figure 3-5 shows the HIPPI-6400 format for a Scheduled Transfer Data Message which originated at the HIPPI-6400 host. Figure 3-6 shows the resulting HIPPI-800 formatted Data message.

Figure 3-5 HIPPI-6400 To HIPPI-800 Data Message
HIPPI-6400 Format
(Fragmenting Translation)



ABC = HIPPI-800 destination address
DEF = HIPPI-6400 source address

Figure 3-6 HIPPI-6400 To HIPPI-800 Data Message
HIPPI-800 Format
(Fragmenting Translation)



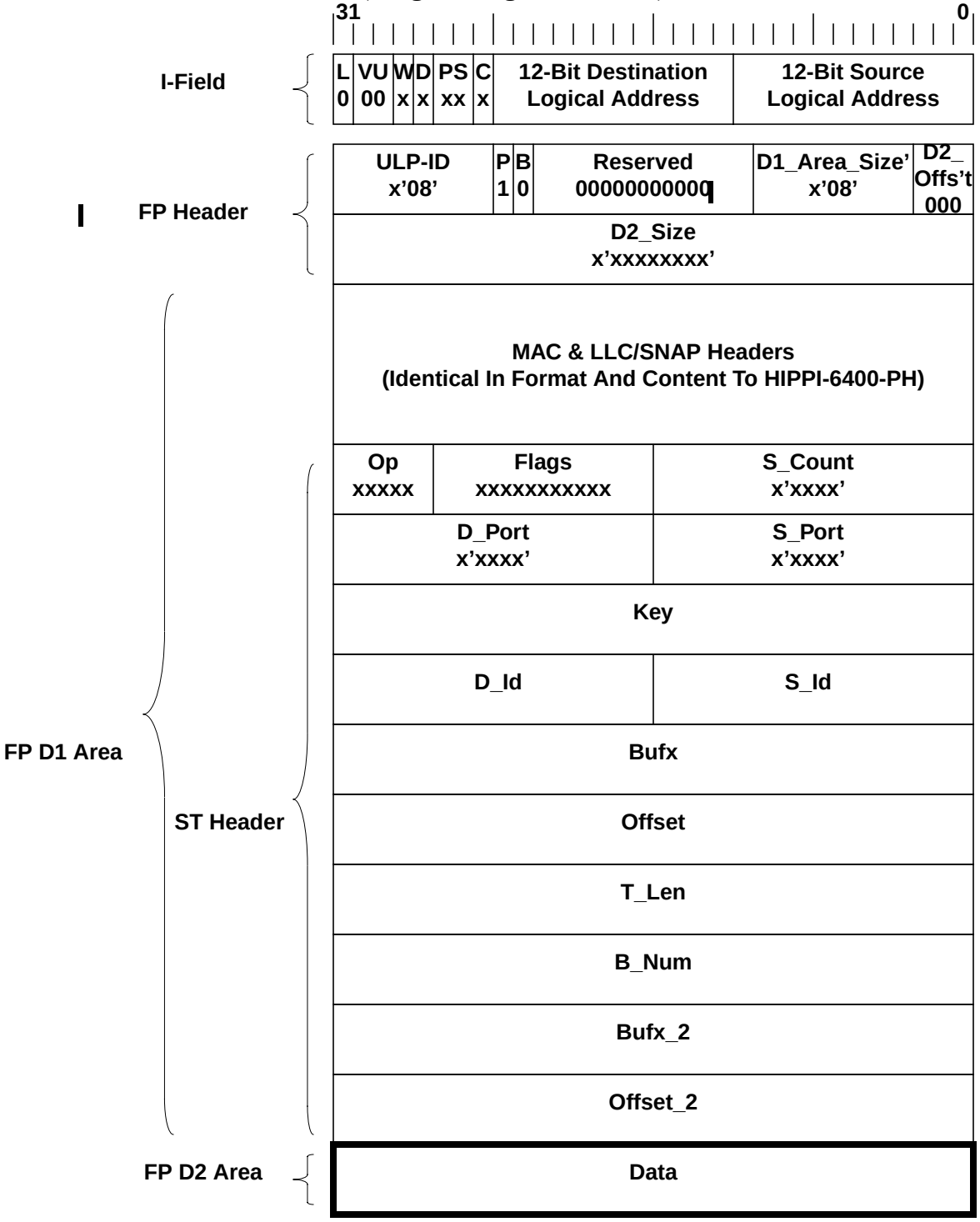
Created by translator

1. **12-Bit Logical Addresses are 12 LSB's of HIPPI-6400 48-bit logical address**
2. **I-Field, FP Header, and D1 Area are created one time only from first HIPPI-6400 Data message. D2 Area is comprised of the data from all HIPPI-6400 STUs (number of STUs set by Replicate value in CTS). HIPPI-6400 MAC & LLC/SNAP headers, and ST headers from all HIPPI-6400 Data messages after the first are discarded.**
3. **D2_Size is Return_D2_Size value from optional payload field of CTS**

3.8 HIPPI-800 To HIPPI-6400 Data Message

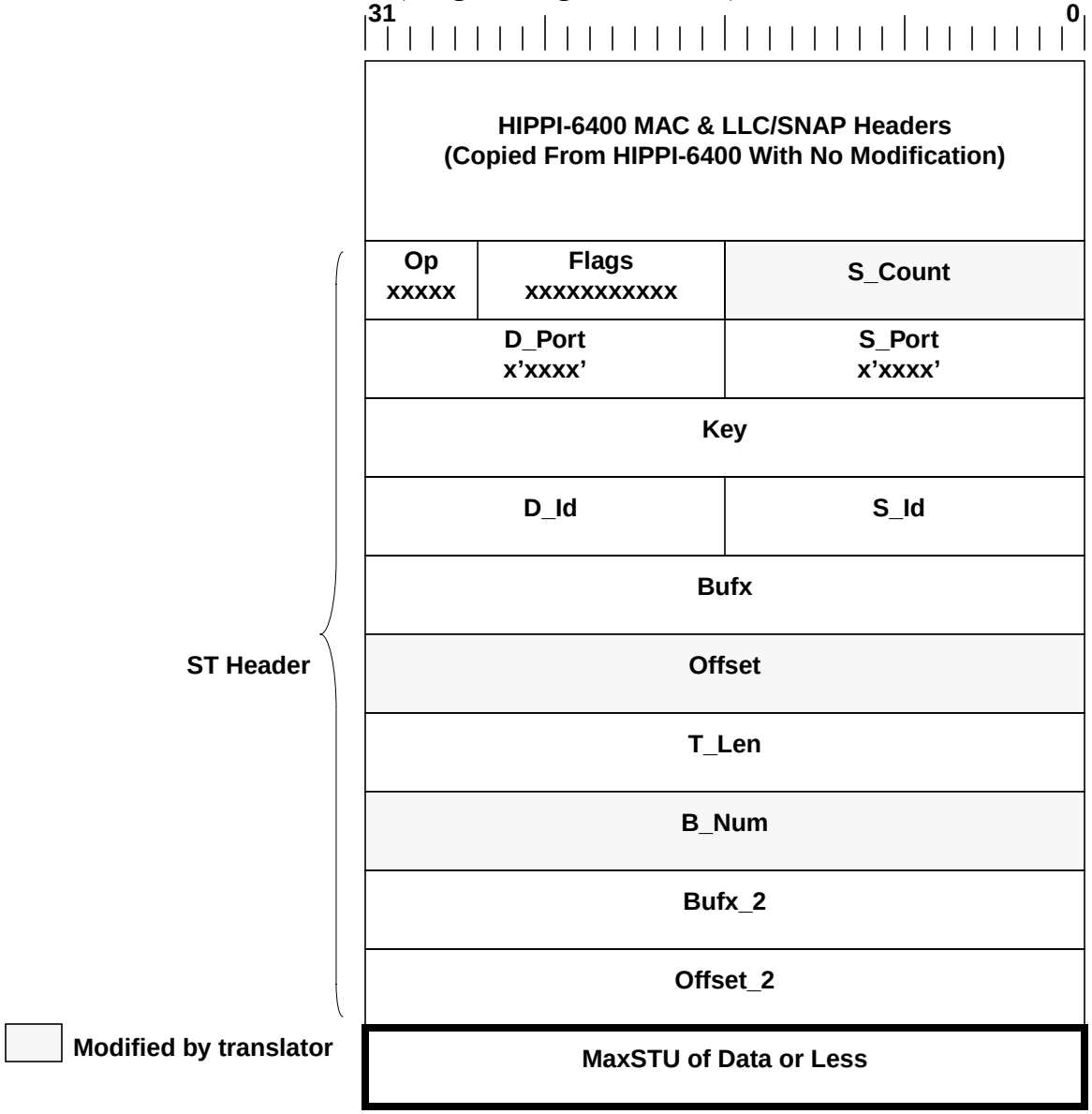
Figure 3-7 shows the HIPPI-800 format for a Scheduled Transfer Data Message which originated at the HIPPI-800 host. Figure 3-8 shows the resulting HIPPI-6400 formatted Data message

Figure 3-7 HIPPI-800 To HIPPI-6400 Data Message
HIPPI-800 Format
(Fragmenting Translation)



1. 12-Bit Destination Logical Address is 12 LSB's of HIPPI-6400 destination host's 48-bit address

Figure 3-8 HIPPI-800 To HIPPI-6400 Data Message HIPPI-6400 Format (Fragmenting Translation)



1. MAC & LLC/SNAP headers and ST header are stored and duplicated for subsequent Data STUs to HIPPI-6400 host (first set sent unmodified)
2. Correct addresses contained in original MAC header from HIPPI-800 D1 Area
3. S_Count starts at value in ST header from HIPPI-800 D1 Area and decrements for each Data STU sent to HIPPI-6400 host
4. B_Num starts at value in ST header from HIPPI-800 D1 Area and increments for each Data STU sent to HIPPI-6400 host
5. Offset starts at value in ST header from HIPPI-800 D1 Area and increments by MaxSTU size for each Data STU sent to HIPPI-6400 host

3.9 END Operation

The HIPPI-800 end shall ensure that no outstanding CTS operations exist before issuing and END or END_ACK operation.

4.0 Considerations

The following restrictions and caveats on operation are needed.

- When using this translation protocol, HIPPI-6400 MaxSTU size shall be a constant fixed size mutually agreed upon by the HIPPI-6400 host, HIPPI-800 host, and the translator. Determination of this size is beyond the scope of this standard. **64KB FOR EIS**
- Bufx and offset values shall be forced "clean" so that the translator does not modify Bufx and simply increments Offset by MaxSTU. Short final STUs shall be supported, short first STUs will not be.
- Block sizes (HIPPI-800 to HIPPI-6400) and the number of combined STUs (HIPPI-6400 to HIPPI-800) should be large enough for efficient operation (>1MB) but small enough so that channel blocking does not reduce bandwidth too much. Channel blocking occurs when the chit-chat to support read operations cannot be sent on the HIPPI-800 channel to the translator because the channel is being used for sending write data. To minimize this problem, write blocks should probably be set to about half of read bundles and the HIPPI-800 driver should give priority to commands on the HIPPI-800 write channel in the following order (highest listed first):
  - scheduling operations
  - writes of command data (i.e. the actual read/write command)
  - writes of data

5.0 Striping Control

If this protocol is used for striping across multiple HIPPI-800 interfaces one RTS does the whole transfer for all of the channel.

With a single RTS, only one end (the HIPPI-800 host) need know about the stripes. Both the HIPPI-800 host and the HIPPI-6400 host must provide at least as many CTS returns as there are stripes. This number of CTS returns should be a multiple of the number of stripes, although this becomes less important as the size of the blocks (or aggregated reads) declines.